[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
G06F 9/455
G06F 12/14

[12] 发明专利申请公开说明书

[21] 申请号 01821575.0

[43] 公开日 2005 年 1 月 5 日

[11] 公开号 CN 1561485A

[22] 申请日 2001.11.27 [21] 申请号 01821575.0
[30] 优先权
[32] 2000.12.27 [33] US [31] 09/752,134
[86] 国际申请 PCT/US2001/045061 2001.11.27
[87] 国际公布 WO2002/052404 英 2002.7.4
[85] 进入国家阶段日期 2003.6.27
[71] 申请人 英特尔公司
地址 美国加利福尼亚州
[72] 发明人 S·仇 G·奈格尔
E·科塔-罗布勒斯 S·耶亚辛
R·乌利 M·科祖克 A·卡吉

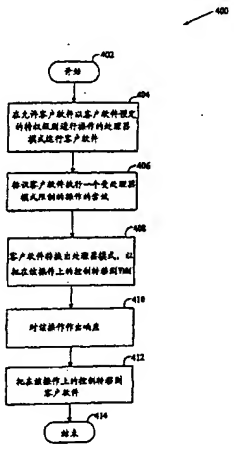
[74] 专利代理机构 中国专利代理(香港)有限公司
代理人 吴立明 王 勇

权利要求书 4 页 说明书 11 页 附图 9 页

[54] 发明名称 用于限制在由虚拟机监控器支持的虚拟机上运行的客户软件操作的新处理器模式

[57] 摘要

在一个实施例中，为客户软件提供了一种处理器模式。处理器模式允许客户软件以由客户软件预定的特权级别进行操作。当客户软件试图执行一个受处理器模式限制的操作时，退出处理器模式，以把在该操作上的控制转移到在这个处理器模式外运行的一个虚拟机监控器。



ISSN1008-4274

知识产权出版社出版

1. 一种方法，包含：

在允许客户软件以由客户软件预定的特权级别进行操作的处理器模式运行客户软件；以及

响应于客户软件执行受所述处理器模式限制的一个操作的尝试，退出所述处理器模式，以把在该操作上的控制转移到在所述处理器模式外运行的 VMM。

2. 如权利要求 1 所述的方法，进一步包含：

对该操作作出响应；以及

通过进入所述处理器模式把在该操作上的控制转移到客户软件。

3. 如权利要求 2 所述的方法，其特征在于：进入所述处理器模式包含加载由客户软件期望的处理器状态。

4. 如权利要求 1 所述的方法，其特征在于：退出所述处理器模式进一步包含：

保存由客户软件使用的处理器状态；以及

加载由 VMM 需要的处理器状态。

5. 如权利要求 1 所述的方法，其特征在于：退出所述处理器模式进一步包含：自动地从与客户软件有关的一个地址空间转移到与 VMM 有关的一个地址空间。

6. 如权利要求 1 所述的方法，进一步包含：在一个处理器控制寄存器中保存一个标志以指示处理器是否在所述处理器模式下。

7. 如权利要求 1 所述的方法，进一步包含：使用被返回在一个处理器寄存器中的多个保留特征位中的一位，来报告处理器支持所述处理器模式的性能。

8. 如权利要求 1 所述的方法，其特征在于：退出所述处理器模式包含：响应于客户软件执行受所述处理器模式限制的操作的尝试，生成多个中断和例外中的一个。

9. 如权利要求 8 所述的方法，其特征在于：生成多个中断和例外中的一个进一步包含：

标识客户软件执行受所述处理器模式限制的操作的尝试；以及

确定客户软件的尝试是可能成功的。

10. 如权利要求 8 所述的方法，进一步包含：

为多个中断和例外保存一个重定向位图，该重定向位图指示多个中断和例外中的每一个是否被允许由客户软件进行处理；以及

参考该重定向位图以确定是否退出所述处理器模式.

11. 如权利要求 8 所述的方法，进一步包含:

标识客户软件以修改一个中断标志的尝试；以及

如果中断标志不控制中断的屏蔽，则修改该中断标志.

12. 如权利要求 8 所述的方法，进一步包含:

标识客户软件以修改一个中断标志的尝试；以及

阻止客户软件修改该中断标志的尝试。

13. 如权利要求 12 所述的方法，其特征在于：阻止客户软件修改该中断标志的尝试包含：为由客户软件进行的修改提供一个遮蔽中断标志。

14. 如权利要求 12 所述的方法，其特征在于：阻止客户软件修改该中断标志的尝试包含：响应于客户软件修改该中断标志的尝试，生成多个中断和例外中的一个。

15. 一个系统，包含:

一个存储器；以及

一个处理器，连接到该存储器，在允许客户软件以由客户软件预定的特权级别进行操作的处理器模式下运行客户软件，标识客户软件执行受所述处理器模式限制的一个操作的尝试，以及响应于该尝试，退出所述处理器模式，以把在该操作上的控制转移到在所述处理器模式外运行的虚拟机器监控器（VMM）.

16. 如权利要求 15 所述的系统，其特征在于：处理器在 VMM 对该操作作出响应之后重新进入所述处理器模式。

17. 如权利要求 16 所述的系统，其特征在于：处理器在重新进入所述处理器模式时将加载由客户软件期望的处理器状态。

18. 如权利要求 15 所述的系统，其特征在于：处理器在退出所述处理器模式时将保存由客户软件使用的处理器状态并且加载由 VMM 需要的处理器状态.

19. 如权利要求 15 所述的系统，其特征在于：退出所述处理器模式进一步包含：自动地从与客户软件有关的一个地址空间转移到与 VMM 有关的一个地址空间.

20. 如权利要求 15 所述的系统，其特征在于：处理器将在一个处理器控制寄存器中保存一个标志以指示处理器是否在所述处理器模式。

21. 如权利要求 15 所述的系统，其特征在于：处理器将使用被返回在一个处理器寄存器中的多个保留特征位中的一位，来报告支持所述处理器模式的性能。

22. 如权利要求 15 所述的系统，其特征在于：响应于客户软件执行受所述处理器模式限制的操作的尝试，处理器将生成多个中断和例外中的一个。

23. 如权利要求 22 所述的系统，其特征在于：在确定客户软件执行受所述处理器模式限制的操作的尝试可能成功时，处理器将生成多个中断和例外中的一个。

24. 如权利要求 22 所述的系统，其特征在于：处理器将参考一个重定向位图以确定是否退出所述处理器模式，该重定向位图指示多个中断和例外中的每一个是否被允许由客户软件进行处理。

25. 如权利要求 22 所述的系统，其特征在于：处理器将标识客户软件修改一个中断标志的尝试，并且如果中断标志不控制中断的屏蔽，则修改该中断标志。

26. 如权利要求 22 所述的系统，其特征在于：处理器将标识客户软件修改一个中断标志的尝试，并且阻止客户软件修改该中断标志的尝试。

27. 如权利要求 26 所述的系统，其特征在于：处理器将通过为由客户软件进行的修改提供一个遮蔽中断标志，来阻止客户软件修改该中断标志的尝试。

28. 一种计算机可读介质，其提供当在一个处理器上执行时导致所述处理器执行以下步骤的指令：

在允许客户软件以由客户软件预定的特权级别进行操作的处理器模式下运行客户软件；以及

响应于客户软件执行受所述处理器模式限制的一个操作的尝试，退出所述处理器模式，以把在该操作上的控制转移到在所述处理器模式外运行的 VMM。

29. 如权利要求 28 所述的计算机可读介质，提供导致处理器执行

以下操作的更多指令：

对该操作作出响应；以及

通过进入所述处理器模式把在该操作上的控制转移到客户软件。

30. 如权利要求 28 所述的计算机可读介质，提供导致处理器执行以下操作的更多指令：

为多个中断和例外保存一个重定向位图，该重定向位图指示多个中断和例外中的每一个是否被允许由客户软件进行处理；以及

参考该重定向位图以确定是否退出所述处理器模式。

用于限制在由虚拟机监控器支持的虚拟机上运行的客户软件操作的新处理器模式

发明领域

本发明通常涉及虚拟机，尤其是涉及提供对虚拟机监控器的处理器支持.

发明背景技术

传统的虚拟机监控器（VMM）通常运行在一台计算机上，并且向其它软件给出一个或多个虚拟机抽象. 每个虚拟机可以起到一个独立平台的作用，运行它自己的“客户操作系统”（即，由 VMM 支持的操作系统）. 客户操作系统期待好象它正在在一台专用计算机而不是虚拟机上运行那样进行操作. 即，客户操作系统希望控制各种计算机操作，并且在这些操作期间可以访问硬件资源. 硬件资源可以包含处理器常驻资源（例如，控制寄存器）和驻留在存储器中的资源（例如，描述符表）. 然而，在虚拟机环境中，VMM 应当能够具有这些资源的最终控制，以提供虚拟机的正确操作和从虚拟机以及在虚拟机之间提供保护. 为了实现这一点，VMM 通常截取和判优由客户操作系统对硬件资源进行的所有访问.

VMMs 的当前实现可以基于用于控制由客户操作系统对硬件资源的访问的软件技术. 然而，这些软件技术可能缺乏阻止客户软件访问在处理器的控制寄存器和存储器中的某些字段的能力. 例如，不能阻止客户操作系统访问在 IA - 32 微处理器的代码段寄存器中的请求者特权级别（RPL）字段. 此外现有的软件技术通常遇到性能问题的困扰. 因此，需要有一种用于支持 VMM 操作的替换机制.

附图简要说明

本发明在附图中通过举例而不是限制进行了说明，在附图中相似的附图标记数字涉及相似的单元，并且其中：

图 1 说明了虚拟机环境的一个实施例；

图 2 说明了基于取消客户特权的虚拟机监控器的操作；

图 3 是依据本发明一个实施例、用于向虚拟机监控器提供处理器支持的系统的方框图；

图 4 是依据本发明一个实施例、用于向虚拟机监控器提供处理器支持的方法的流程图；

图 5 是依据本发明一个实施例、用于执行转换出 V32 模式的方法的流程图；

图 6 是依据本发明一个实施例、用于生成虚拟化陷阱的方法的流程图；

图 7 是依据本发明一个实施例、用于维持一个重定向映射的方法的流程图；

图 8 是依据本发明一个实施例、用于控制中断屏蔽的方法的流程图；以及

图 9 是处理系统的一个实施例的方框图。

实施例描述

描述了一种用于向虚拟机监控器提供处理器支持的方法和装置。在下面的描述中，为了说明起见，阐述了大量细节以便提供对本发明的一个彻底了解。然而，显然对本领域普通技术人员来说没有这些特定的细节本发明也能够被实现。

依据在计算机存储器内在数据位上的操作的算法和符号表示给出随后的某些部分详细说明。这些算法描述和表示是由数据处理领域内技术人员使用的、以最有效地向该领域其它技术人员传送它们的工作实质的手段。在此，并且通常，算法被认为是导致一个期望结果的自相容的一系列步骤。这些步骤是要求物理量的物理操作的那些步骤。通常，尽管不一定，这些量采取能够被存储、传送、组合、比较、及其它处理的电或者磁信号的形式。有时，主要由于公共使用的原因，已经证明把这些信号称为比特、值、单元、符号、字符、术语、数字等是方便的。

然而，应当记住：所有这些和类似的术语与适当的物理量有关，并且仅仅是用于这些量的方便的标记。除非特别地说明，否则如从以下讨论中明显看出的那样，应当理解，在整个本发明中，利用术语诸如“处理”或者“计算”或者“确定”或者“显示”等可以涉及一个计算机系统或者类似电子计算设备的动作和处理过程，其把在计算机系统的寄存器和存储器内被表示为物理（电子）量的数据处理和变换成为类似地在计算机系统存储器或者寄存器或者其它这种信息存储

器、传输或者显示设备内被表示为物理量的其它数据.

本发明在此还涉及用于执行这些操作的装置. 这个装置可以为需要的目的而被特别地构造，或者它可以包含一个由保存在计算机中的计算机程序有选择地激活或者重新配置的通用计算机. 这种计算机程序可以被保存在计算机可读存储介质中，该计算机可读存储介质诸如但是不局限于：包含软盘、光盘、CD－ROMs、和磁光盘的任何类型的磁盘、只读存储器(ROMs)、随机存取存储器(RAMs)、EPROMs、EEPROMs、磁或者光卡、或者适于存储电子指令的任何类型的介质，而且上述每个都连接到一条计算机系统总线. 指令是可使用一个或多个处理设备（例如，处理器、中央处理器等）执行的.

在此给出的算法和显示不是固有地与任何特定计算机或者其它装置有关. 可以使用具有依据在此的示教的程序的各种通用机，或者可以证明构造更多专用装置以执行所需要的方法步骤是方便的. 从以下的描述中将会呈现用于各种这些机器的所需要的结构. 此外，本发明不是参照任何特定程序设计语言进行描述的. 应当理解，可以使用各种程序设计语言来实现在此描述的本发明的示教.

在以下实施例的详细说明中，参考举例显示了其中可以实现本发明的特定实施例的附图. 在这些附图中，在几个视图当中相似的数字基本上地描述了类似的元件. 足够详细地描述了这些实施例以使本领域技术人员能实现本发明. 可以使用其它实施例，并且可以进行结构上的、逻辑的、和电学的变化，而没有背离本发明的范围. 此外，应当理解，本发明的各个实施例尽管是不同的但未必是互相排斥的. 例如，在一个实施例中描述的一个特定特征、结构、或者特性可以被包含在其它实施例中. 因此，下列详细说明不是限制的含义，而且本发明的范围仅仅由附加权利要求、以及这种权利要求被授权的整个等效范围来定义.

本发明中的方法和装置为虚拟机监控器（VMM）提供处理器支持. 图 1 说明了其中本发明可以实现的虚拟机环境 100 的一个实施例. 在这个实施例中，裸平台硬件 116 包含一个计算平台，其能够例如执行标准的操作系统（OS）或者虚拟机监控器（VMM）、诸如 VMM 112. 通常以软件实现的 VMM 可以向高层软件输出一个裸机接口、诸如模拟器. 这种高层软件可以包含标准或者实时 OS，尽管本发明在这方面在范围

上不受限制，并且做为选择，例如，VMM 可以在另一个 VMM 内或者在另一个 VMM 上面执行. VMMs 和它们的典型特征和功能对于本领域技术人员来说是公知的，并且可以例如以软件、固件或者各种技术的组合来实现.

如上所述，VMM 向其它软件（即，“客户”软件）给出一个或多个虚拟机（VMs）抽象. 图 1 显示了两个 VMs 102 和 114. 每个 VM 的客户软件包含一个客户 OS，诸如客户 OS 104 或者 106，以及各种客户软件应用程序 108 - 110. 每一个客户 OSs 104 和 106 希望控制对在正在其上运行客户 OS 104 或者 106 的硬件平台内的物理资源（例如，处理器寄存器、存储器和存储器映射的 I/O 设备）的访问以及执行其它功能. 然而，在虚拟机环境中，VMM 112 应当具有在物理资源上的最终控制，以提供 VMs 102 和 112 的正确操作和从 VMs 102 和 114 以及在 VMs 102 和 114 之间提供保护. VMM 112 通过截取客户 OSs 104 和 106 对计算机的物理资源的所有访问来实现这个目的.可以使用各种技术以允许 VMM 112 截取上述访问. 这种技术中的一种是取消客户特权技术，它迫使所有的客户软件以不允许软件访问某些硬件资源的硬件特权级别运行. 因此，每当客户 OS 104 或者 106 试图访问这些硬件资源中的任何一个时，它“设置陷阱”到 VMM 112，即，如果由客户 OS 启动的一个操作涉及访问这种硬件资源则 VMM 112 接收在这个由其启动的操作上的控制.

图 2 说明了支持取消客户特权的 VMM 的操作的现有技术实施例. 如上所述，取消客户特权迫使客户 OS 在较少的执行特许方式下执行. 就 IA - 32 微处理器来说，基于页面的保护的特性是使所有的客户软件以最小特权级别（即环 3）运行. 即，客户 OS 206 和客户应用程序 204 以同样的特权级别运行. 因此，客户 OS 206 不能从客户应用程序 206 中保护它自己，由此可能损害客户 OS 206 的完整性. 这个问题被称为环压缩.

取消客户特权还可能导致地址空间压缩问题. 如上所述，客户软件访问硬件资源的某些尝试导致转移控制到 VMM 220 的陷阱. 为了允许这个控制转移，一部分 VMM 代码和/或数据结构在体系结构上可以被要求驻留在与客户 OS 206 相同的虚拟地址空间中. 例如，IA - 32 指令集体系结构（ISA）可以要求中断描述符表（IDT）212、全局描述符

表（GDT）210 和陷阱处理例程驻留在与客户 OS 206 相同的虚拟空间中．必须保护驻留在虚拟空间 202 中的 VMM 代码和数据结构 220 不能由客户软件（例如，通过在环 0 运行）进行访问．因此，客户 OS 206 没有如客户 OS 206 期望的那样控制整个地址空间 202．这导致地址空间压缩问题．

使用取消客户特权的 VMMs 的另一个限制适合于其中处理器未能阻止客户软件读取有特权的硬件资源的某些情况．例如，IA － 32 微处理器允许客户 OS 206 执行 PUSH CS 指令，其把一个代码段寄存器存储到存储器中．这个寄存器字段中的一个字段存储有关当前特权级别的信息．因此，客户 OS 206 能够通过从存储器中读取当前特权级别的值，知道它的特权级别是 3，而不是如客户 OS 206 期望的那样是 0．因此，客户 OS 206 可能暴露出这个事实：它正在一个虚拟机上运行，并且客户 OS 206 的完整性可以被损害．

类似地，在某些情况下，处理器没有对客户软件的尝试设置陷阱以修改有特权的软件资源．例如，IA － 32 处理器允许客户 OS 206 发出试图加载 EFLAGS 的 POPF 指令，而不是生成一个陷阱，而且简单地忽略客户 OS 206 的所有或者部分的这种尝试，这是因为客户 OS 206 用不够的特权执行这些指令．因此，客户 OS 206 相信相应的一个 EFLAGS 字段已经被修改了，但是 VMM 220 不知道那个情况并且不能适当地模拟这个修改．因此，客户 OS 206 可能暴露出这个事实：它正在一个虚拟机上运行，并且客户 OS 206 的完整性可以被损害．

使用取消客户特权的 VM 监控器的另一个限制是由过度设置陷阱所引起的．由于需要保护不被客户软件访问的硬件资源单元的数目是有效的而且这种访问可能是频繁的，所以陷阱经常发生．例如，IA － 32 微处理器支持 CLI 指令．发布 CLI 指令以修改中断标志，该中断标志是有特权的硬件资源的一个单元，并且因此不能由无特权的软件访问．客户 OS 206 通常在它的操作期间发布这些指令，由此导致频繁地到 VMM 220 的陷阱．频繁地设置陷阱反面地影响了系统性能，并且减少了 VMM 220 的实用性．

本发明通过为 VMM 提供处理器支持解决了上述问题和各种其它限制．图 3 是依据本发明一个实施例、用于向虚拟机监控器提供处理器支持的系统的方框图．

参见图 3，所有的客户软件在在此被称为虚拟 32 位模式（V32 模式）的处理器模式下运行。V32 模式允许客户软件以它的预定特权级别运行。例如，就 IA－32 ISA 来说，客户 OS 308 以最高特权级别（即，环 0）运行，而客户应用程序 306 以最低特权级别（即，环 3）运行。V32 模式通过阻止客户软件执行可能导致它访问某些有特权的硬件资源的操作来限制客户软件的操作。当客户软件试图执行这样一个操作时退出 V32 模式。

VMM 320 在 V32 模式外运行。当发生转换出 V32 模式时，VMM 320 接收在由客户 OS 308 或者客户应用程序 306 启动的操作上的控制。VMM 320 然后执行这个操作，并且通过进入 V32 模式把控制转移回客户软件，借此模拟客户软件期望的功能。

在一个实施例中，通过在处理器的一个控制寄存器（例如 CR0）中保存一个标志以指示处理器是否在 V32 模式下来实现 V32 模式。在另一个实施例中，这个标志（在此被称为 EFLAGS.V32）被保存在 EFLAGS 的上半部分中的一个保留位中。EFLAGS.V32 标志通过从 V32 模式中转换出或者转换成为 V32 模式而被修改。

在一个实施例中，使用保留的特征位中的一位报告处理器支持 V32 模式的性能，其中被保留的特征位当在 EAX 中用值 1 执行 CPUID 指令时在 EDX 中被返回。应当注意到，能够使用其它各种机制来实现 V32 模式和报告处理器支持 V32 模式的性能，而不影响通用性。

在一个实施例中，某些例外和中断导致转换出 V32 模式。这些例外和中断包含“虚拟化陷阱”。虚拟化陷阱是当在 V32 模式下运行的客户软件试图执行一个可能导致它访问某些有特权的硬件资源的操作时生成的。在一个实施例中，当发生转换出 V32 模式时，客户地址空间 304 被自动地变换成 VMM 地址空间 302。此外，由客户软件使用的处理器状态被保存在暂时寄存器中，并且加载由 VMM 320 需要的处理器状态。

在一个实施例中，当发生到 V32 模式的转换时，在转换出 V32 模式（即到 VMM 320）时所保存的处理器状态被自动地恢复，VMM 地址空间 302 被变换成客户地址空间 304，并且把控制返回给客户 OS 308。

在一个实施例中，当客户软件在 V32 模式下运行时，由客户 OS 308 使用客户 IDT（即，驻留在客户地址空间 304 中的 IDT）处理软件中断

（例如，由执行 BOUND、INT 或者 INTO 指令所引起的中断）．包含虚拟化陷阱的其它所有中断和例外导致转换出 V32 模式，其导致客户地址空间 304 变换到 VMM 地址空间 302．然后使用 IDT 316 以指向处理相应的一个例外或者中断的代码．

在一个实施例中，保存一个新的中断标志（即，虚拟机中断标志）用于由客户软件访问．每当客户软件试图访问中断标志（IF）时，作为替代它将访问虚拟机中断标志（VMIF）．在一个实施例中，除了当客户 OS 308 刚好设置 VMIF 为 1（例如，通过 STI 指令）和 VMM 320 希望向客户 OS 308 传送一个待处理中断时以外，客户软件访问 VMIF（例如，使用 CLI 指令）的尝试不会导致转换出 V32 模式．在此被称为“虚拟待处理中断”的这种待处理中断生成虚拟化陷阱，其允许 VMM 320 在客户 OS 308 发信号指示它准备好处理这样一个中断时向客户软件传送一个待处理中断．在一个实施例中，在 EFLAGS 寄存器的上半部分中的保留位中的一个被用来保存一个指示客户软件是否具有一个待处理虚拟中断的标志．

V32 模式的实现允许解决如上所述的、取消客户特权导致的所有问题．特别地，由于客户软件在 V32 模式下以它预定的特权级别运行，所以消除了环压缩的问题．此外，由于虚拟化陷阱自动导致到 VMM 地址空间 302 的切换，所以地址空间压缩不再是一个问题，并且因此既不要求在客户地址空间 304 中驻留控制这种传送的表格，也不要求在客户地址空间 304 中驻留处理相应的一个虚拟化陷阱的代码．

此外，由于 V32 模式允许客户软件以它的预定特权级别运行，所以需要被保护的硬件资源不再包含控制特权级别的硬件资源的那些单元．例如，由于存储有关当前特权级别的信息的代码段寄存器中的字段现在存储由客户 OS 308 预定的特权级别，所以以上所述的 PUSH CS 指令不能再向客户 OS 308 揭示它在一台虚拟机上运行．类似地，由于客户 OS 206 用足够的特权执行这些指令，所以试图加载 EFLAGS 的 POPF 指令在由客户 OS 308 执行时不再被忽略．

因此，需要被保护的硬件资源的单元数目减少了．如果它们中的任何一个允许客户软件的未设置陷阱的读或者写访问，则它们被特别地设计以当在 V32 模式下执行时导致陷阱．因此，消除了由未设置陷阱的读和写访问所引起的问题．此外，由于 V32 模式的实现减少了需

要被保护的硬件资源的单元数目，所以当客户软件试图访问这些单元时发生的陷阱数目也减少了．此外通过提供用于消除由最常使用的指令所引起的陷阱的机制，减少了陷阱的频率．例如，除了当客户软件具有一个待处理的虚拟中断时以外，STI 指令不再导致陷阱．

图 4 是依据本发明一个实施例、用于向虚拟机监控器提供处理器支持的方法 400 的流程图．在处理块 404，在处理器模式（即 V32 模式）下执行客户软件，该处理器模式允许客户软件以由客户软件预定的特权级别进行操作．即，客户 OS 可以以管理员特权级别进行操作，而客户应用程序可以以用户特权级别进行操作．

在处理块 406，标识客户软件执行受 V32 模式限制的一个操作的尝试．响应于这个尝试，退出 V32 模式以转移在由客户软件启动的操作上的控制到在 V32 模式外运行的 VMM（处理器块 408）．在一个实施例中，VMM 设定什么操作应当导致转换出 V32 模式，如以下连同图 7 更详细描述的那样．在一个实施例中，这种操作生成导致转换出 V32 模式的虚拟化陷阱．做为选择，能够使用在本领域已知的其它任何机制以导致转换出 V32 模式．以下连同图 5 更详细地描述执行转换出 V32 模式的一个实施例．

此外，VMM 对由客户软件预定的操作作出响应（处理块 410）．然后，重新进入 V32 模式以把在这个操作上的控制转移回客户软件（处理块 412），并且方法 400 返回给处理块 404．在一个实施例中，当发生到 V32 模式中的转换时，由客户软件期望的处理器状态被自动地恢复，并且 VMM 地址空间被变换到客户地址空间．

图 5 是依据本发明一个实施例、用于执行转换出 V32 模式的方法 500 的流程图．方法 500 随保存由客户软件使用的处理器状态开始（处理块 504）．在一个实施例中，保存的处理器状态被存储在处理器的暂时寄存器中．在处理块 506，由 VMM 需要的处理器状态被加载到处理器寄存器中．在一个实施例中，加载处理器状态影响客户地址空间到 VMM 地址空间的变换（例如，通过加载控制寄存器 CR3 来加载处理器状态）．在一个替换实施例中，加载处理器状态不会导致在地址空间中的变换．在这样一个实施例中，在处理块 508，执行地址空间切换以从客户地址空间转移到 VMM 地址空间．因此，当发生一个导致转换的中断或者例外时，驻留在 VMM 地址空间中的 IDT 被自动地用来指向用于处理

这个中断或者例外的 VMM 常驻代码.

图 6 是依据本发明一个实施例、用于生成虚拟化陷阱的方法 600 的流程图. 方法 600 随标识客户软件执行一个可能受 V32 模式限制的操作的尝试开始（处理块 604）. 在判定框 606，就客户软件的尝试是否可能成功进行确定. 如果确定是肯定的，则生成一个虚拟化陷阱（处理块 608）. 或者，不生成虚拟化陷阱，并且客户软件继续进行该操作（处理块 610）. 例如，依据 IA - 32 ISA，RDMSR 指令仅仅能够由以管理员特权运行的软件执行. 因此，如果用管理员特权运行的客户软件 OS 执行这条指令，则它的尝试可能是成功的. 如果用用户特权运行的一个客户应用程序执行这条指令，则它的尝试将不会成功，并且将发生一个一般保护故障. 因此，客户 OS 执行 RDMSR 指令的尝试将导致一个虚拟化陷阱，但是客户应用程序的尝试将由客户 OS 进行处理.

在一个实施例中，虚拟化陷阱将由客户 OS 访问处理器的控制寄存器（例如 CR0 - CR4）的可能成功的尝试引起. 例如，就 IA - 32 处理器来说，将响应于客户软件执行 MOV CR（除存储 CR2 的尝试之外，其不需要导致虚拟化陷阱）、CLTS、LMSW 或者 SMSW 指令、或者任务切换的尝试而生成虚拟化陷阱. 如果客户软件具有一个待处理的虚拟中断，则虚拟化陷阱也可能由客户软件设置一个中断标志 IF（例如，经由 STI、POPF 或者 IRET 指令）的可能成功的尝试所引起. 就 IA - 32 ISA 来说，执行这种指令、例如 HLT、IN、INS / INSB / INSW / INSD、INVD、OUT、OUTS / OUTSB / OUTSW / OUTSD、RDMSR、和 WRMSR 的成功尝试将导致虚拟化陷阱. 这些虚拟化陷阱将阻止客户软件异常中止处理器和直接访问 I / O 端口、高速缓存或者特定模式的寄存器. 此外，虚拟化陷阱可能由以下尝试所引起：执行 CPUID 指令以允许 VMM 给出由 VMM 选择的处理器抽象特征的尝试、执行 INVLPG 指令以允许 VMM 适当地虚拟化地址转换的尝试、以及执行由客户软件使用的 IRET 指令（如果 IRET 被用来转换到 V32 模式中）以实现一个 VMM 以便允许递归嵌套的 VMMs 的尝试.

图 7 是依据本发明一个实施例，用于维持一个重定向映射的方法 700 的流程图. 依据这个实施例，VMM 保存一个重定向映射以设定哪些中断和例外应当导致一次虚拟化陷阱（处理块 704）. 在处理块 706，标识一个中断或者例外的发生. 然后参考该重定向映射，以在重定向

位图中找到与这个中断或者例外有关的一位（处理块 708）。

在判定框 710，就这个中断是否被允许由客户 OS 处理进行确定。如果确定是肯定的，则该中断或者例外被传送到 V32 模式，并且由客户 OS 进行处理（处理块 714）。或者，生成一个虚拟化陷阱，导致转换出 V32 模式（处理块 712）。

图 8 是依据本发明一个实施例，用于控制中断屏蔽的方法 800 的流程图。可以使用各种实施例来控制中断的屏蔽。在一个实施例中，当客户软件正在执行时，不屏蔽所有的中断。在这个实施例中，客户软件被允许对一个中断标志（例如，就 IA - 32 微处理器来说，这个标志被标识为 EFLAGS.IF）进行处理，但是相对于中断的屏蔽来说这个处理将被忽略。在另一个实施例中，中断的屏蔽取决于该中断标志。在这个实施例中，不允许客户软件处理该中断标志。特别地，可以通过为由客户软件进行的修改提供一个遮蔽中断标志（例如 EFLAGS.VMIF）、通过响应于客户软件的这样一个尝试生成一个虚拟化陷阱、或者通过使用在本领域已知的其它任何技术，来阻止客户软件访问该中断标志。

方法 800 随标识客户软件修改一个可能控制中断屏蔽的中断标志的尝试开始（处理块 804）。在判定框 806，就中断标志是否控制中断的屏蔽进行确定。如果确定是否定的，即没有屏蔽所有的中断，则允许客户软件修改该中断标志（处理块 808）。如上所述，这个修改不会对中断的屏蔽有影响。

否则，如果中断的屏蔽取决于该中断标志，则就是否存在一个遮蔽中断标志、即客户软件影响中断屏蔽的尝试是否正在影响遮蔽标志进行确定（判定框 810）。如果确定是否定的，即客户软件试图修改实际的中断标志，则发生一个虚拟化陷阱（处理块 812），导致转换出 V32 模式（处理块 816）。做为选择，如果实际的中断标志不能由客户软件进行访问，则允许客户软件修改该遮蔽中断标志（处理块 814）。

图 9 是处理系统的一个实施例的方框图。处理系统 900 包含处理器 920 和存储器 930。处理器 920 能够是能执行软件的任何类型的处理器，诸如微处理器、数字信号处理器、微控制器等。处理系统 900 能够是个人计算机（PC）、大型机、手持设备、便携式计算机、置顶盒、或者包含软件的其它任何系统。

存储器930能够是硬盘、软盘、随机存取存储器（RAM）、只读存储器（ROM）、闪速存储器、或者可由处理器920读取的其它任何类型的机器介质。存储器930能够存储用于执行实现本发明的各种方法实施例、诸如方法400、500、600、700和800（图4－8）的指令。

应当理解，以上描述是用于说明性的，而不是限制。对本领域技术人员来说在阅读和理解上述说明书后许多其它实施例将会是显然的。本发明的范围因此应当参考附加的权利要求、以及这种权利要求被授权的整个等效范围来确定。

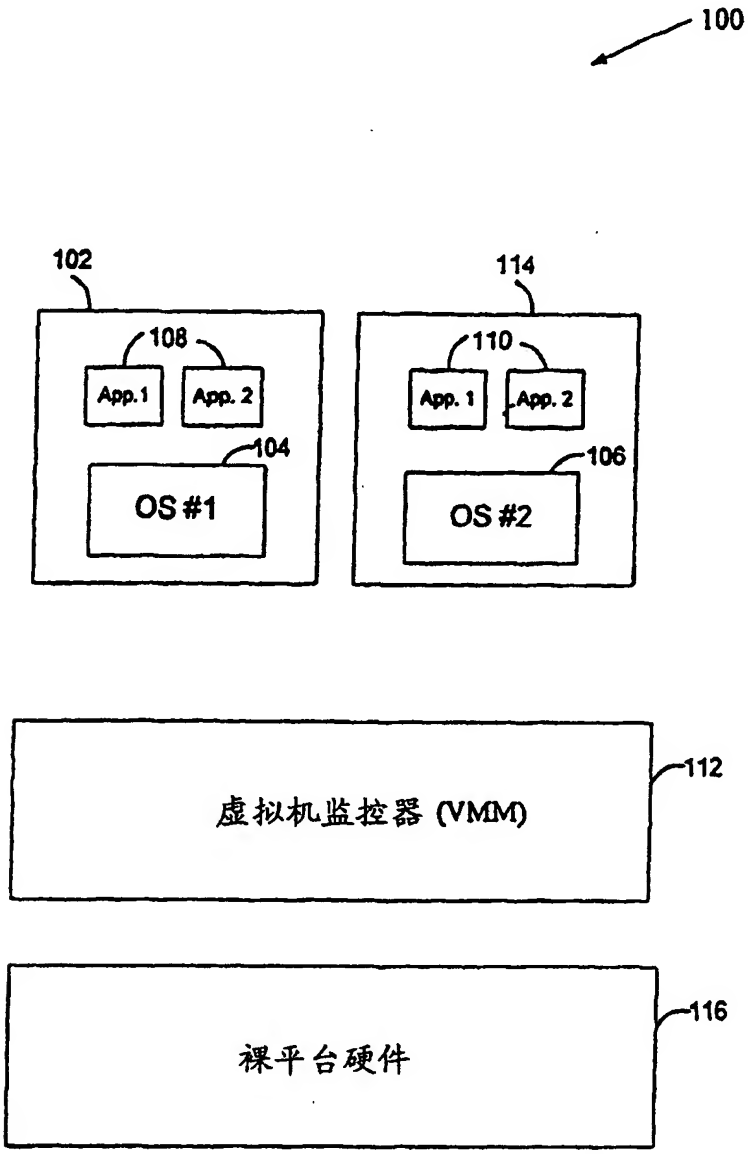

100
102
114
108
App.1
App. 2
104
OS #1
110
App. 1
App. 2
106
OS #2
虚拟机监控器 (VMM)
112
裸平台硬件
116

图 1

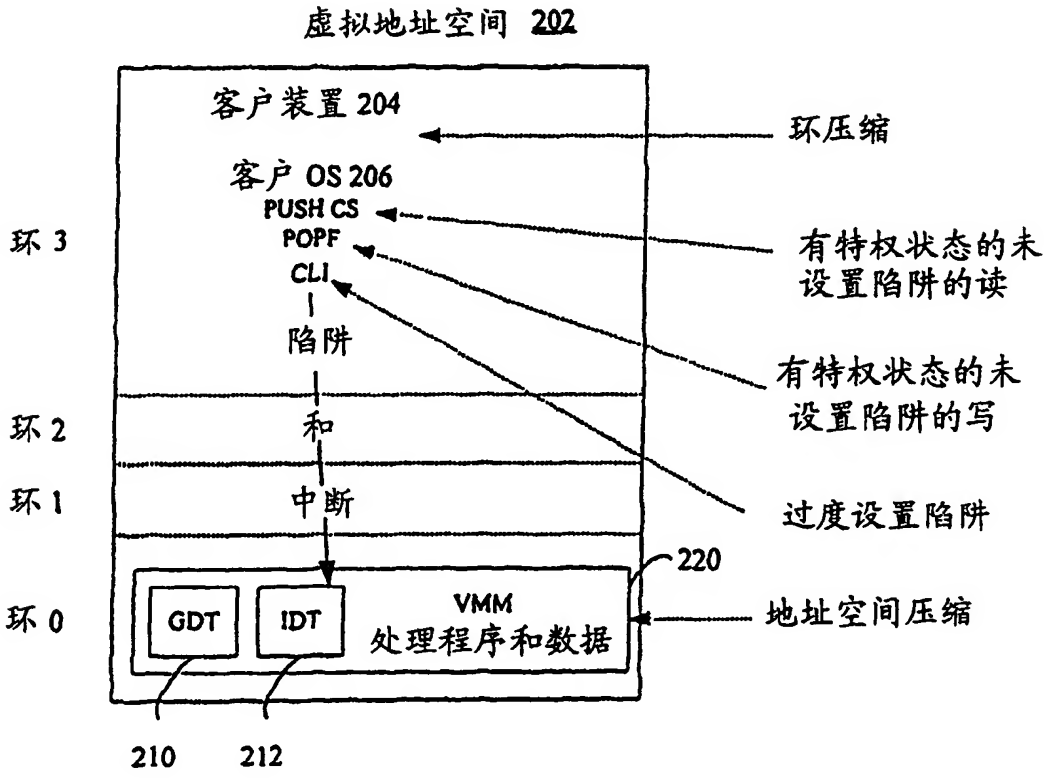

虚拟地址空间 202
客户装置 204
环压缩
客户 OS 206
PUSH CS
POPF
CLI
环 3
有特权状态的未设置陷阱的读
陷阱
有特权状态的未设置陷阱的写
环 2
和
环 1
中断
过度设置陷阱
220
环 0
GDT
IDT
VMM
处理程序和数据
地址空间压缩
210
212

图 2

现有技术

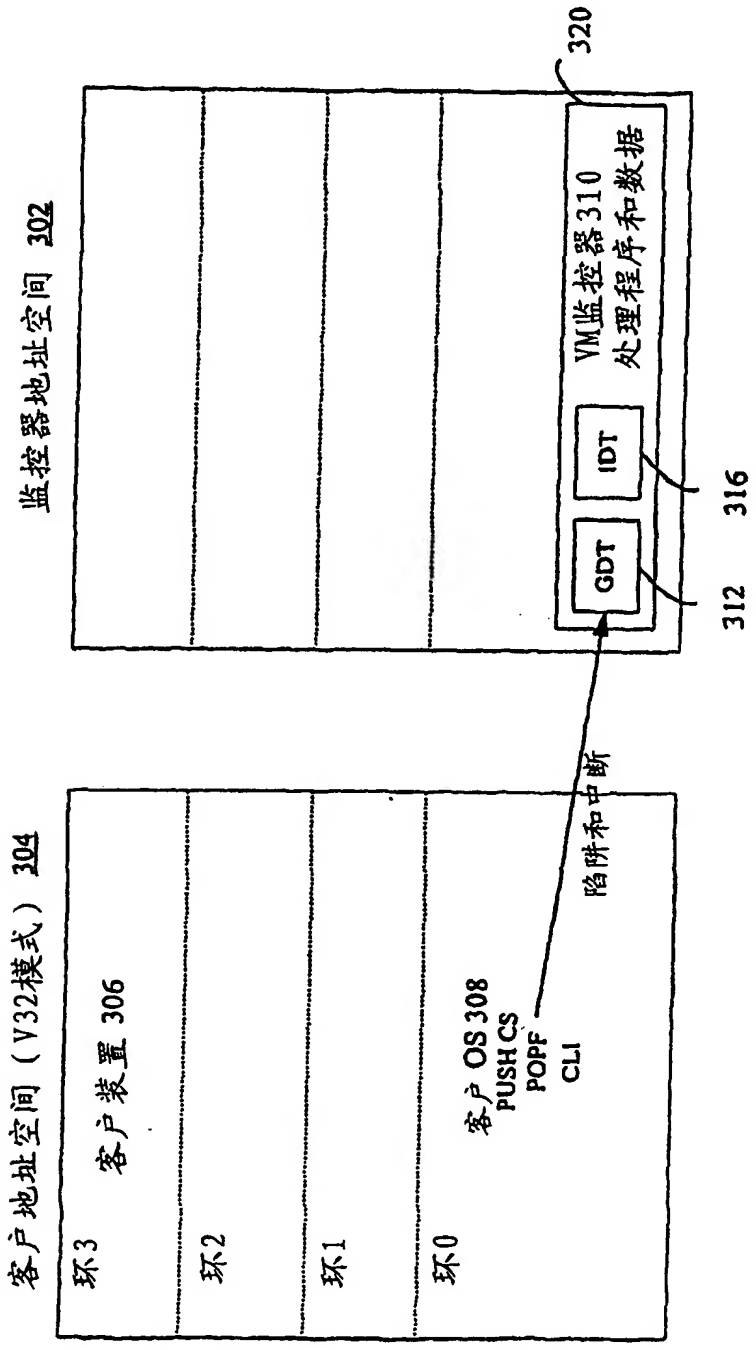

监控器地址空间 302
320
VM监控器310
处理程序和数据
IDT
GDT
312
316
客户地址空间（V32模式） 304
环3
客户装置 306
环2
环1
环0
客户 OS 308
PUSH CS
POPF
CLI
陷阱和中断

图 3

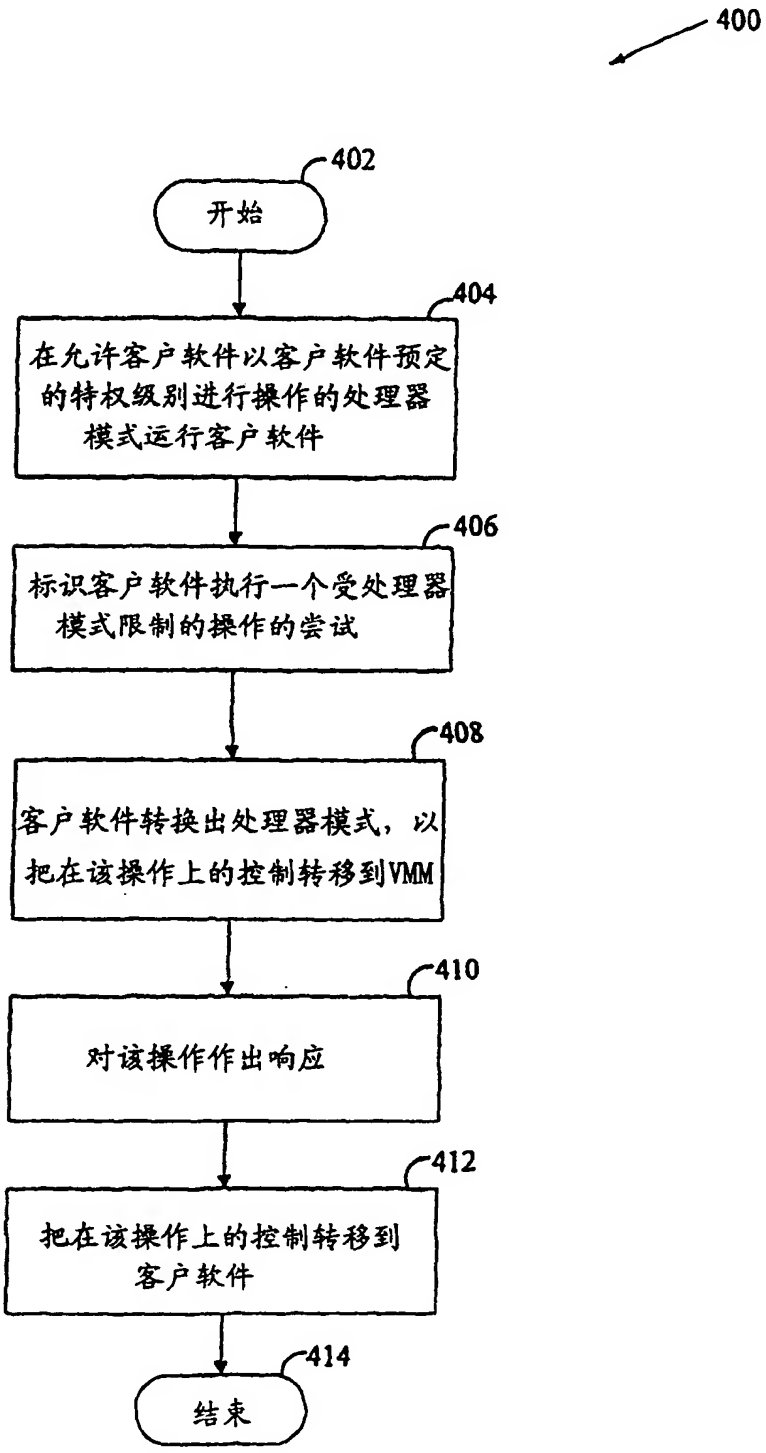

400
402
开始
404
在允许客户软件以客户软件预定的特权级别进行操作的处理器模式运行客户软件
406
标识客户软件执行一个受处理器模式限制的操作的尝试
408
客户软件转换出处理器模式，以把在该操作上的控制转移到VMM
410
对该操作作出响应
412
把在该操作上的控制转移到客户软件
414
结束

图 4

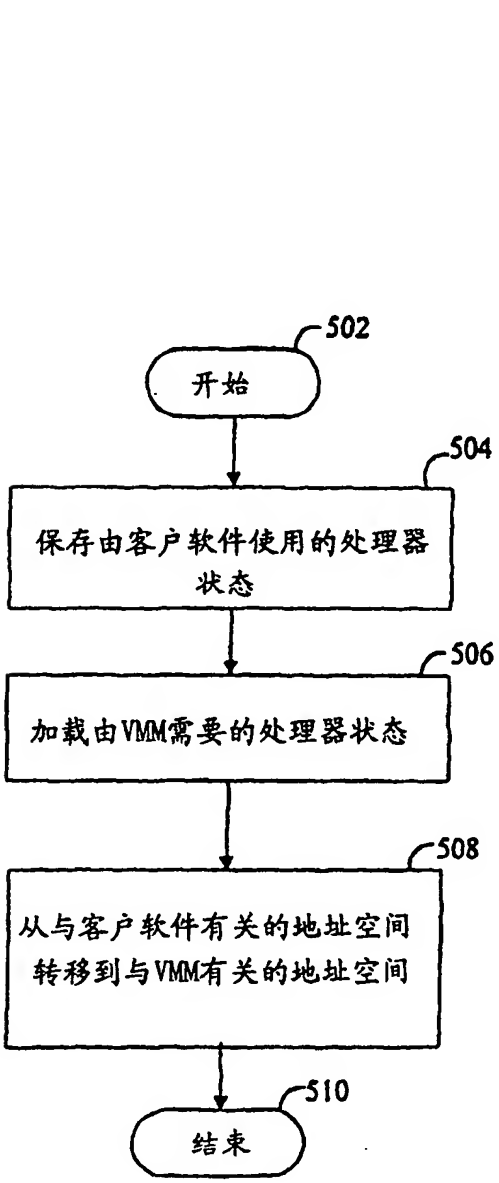

500
502
开始
504
保存由客户软件使用的处理器状态
506
加载由VMM需要的处理器状态
508
从与客户软件有关的地址空间转移到与VMM有关的地址空间
510
结束

图 5

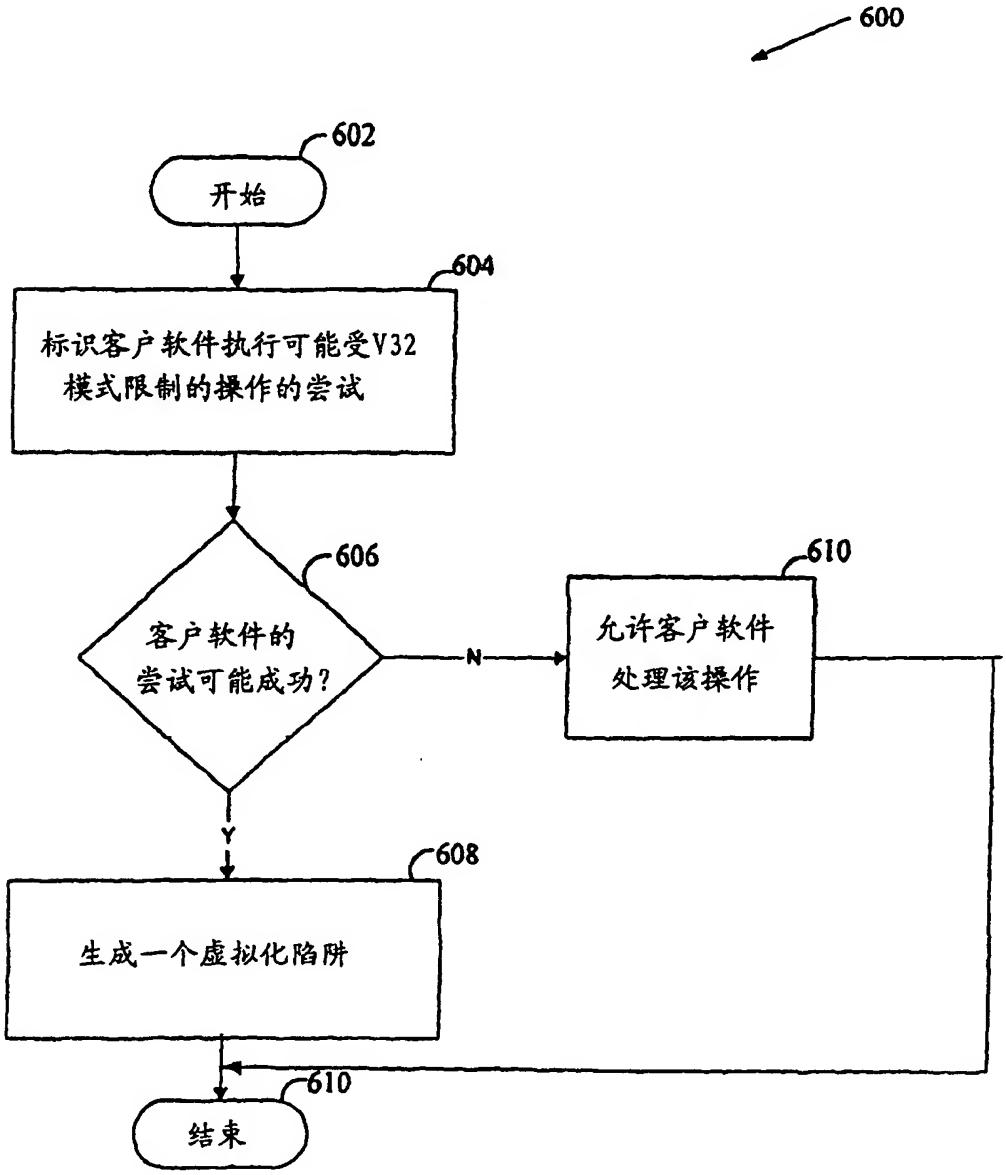

600
602
开始
604
标识客户软件执行可能受V32模式限制的操作的尝试
606
客户软件的尝试可能成功?
N
610
允许客户软件处理该操作
Y
608
生成一个虚拟化陷阱
610
结束

图 6

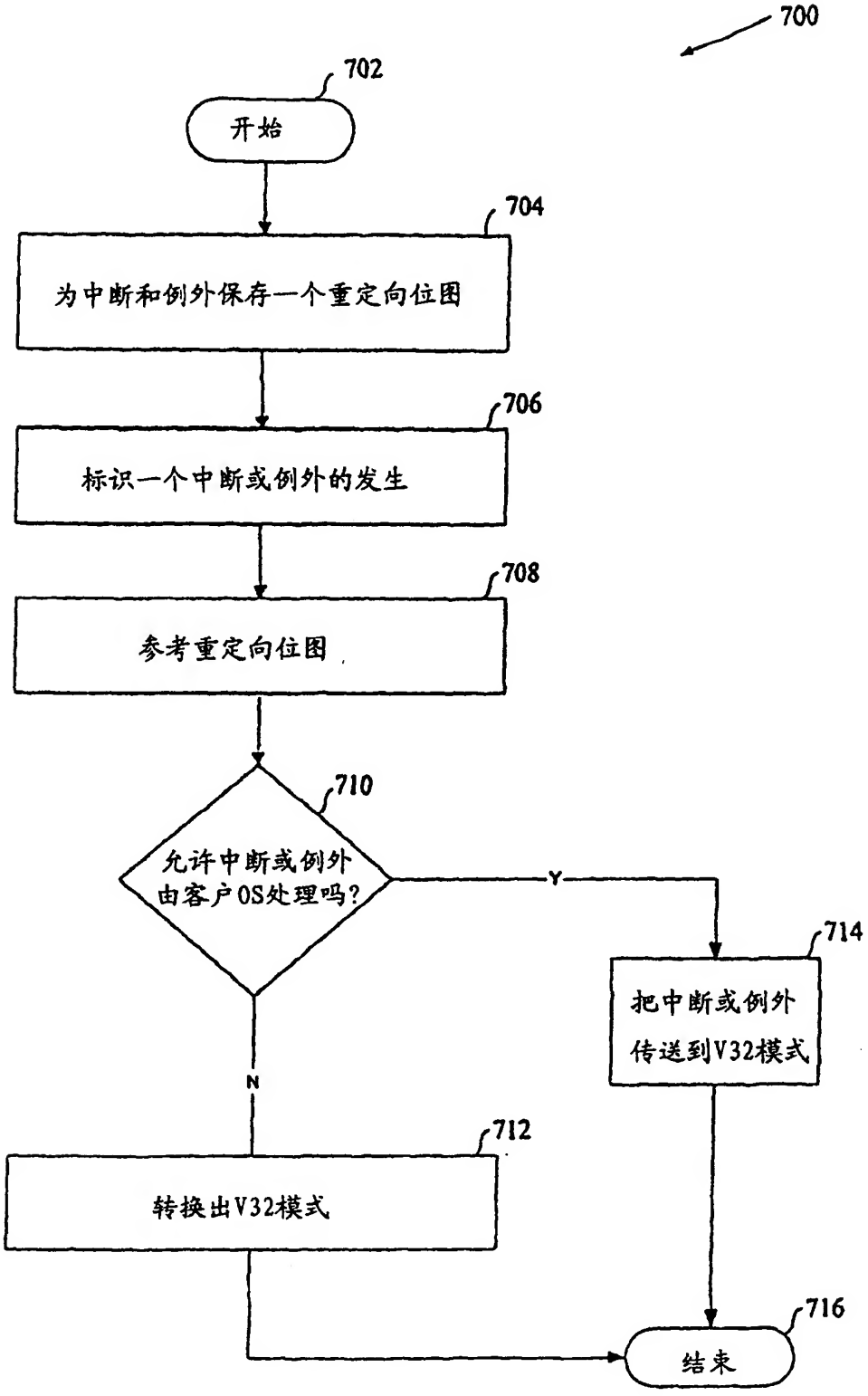

700
702
开始
704
为中断和例外保存一个重定向位图
706
标识一个中断或例外的发生
708
参考重定向位图
710
允许中断或例外
由客户OS处理吗?
Y
N
714
把中断或例外
传送到V32模式
712
转换出V32模式
716
结束

图 7

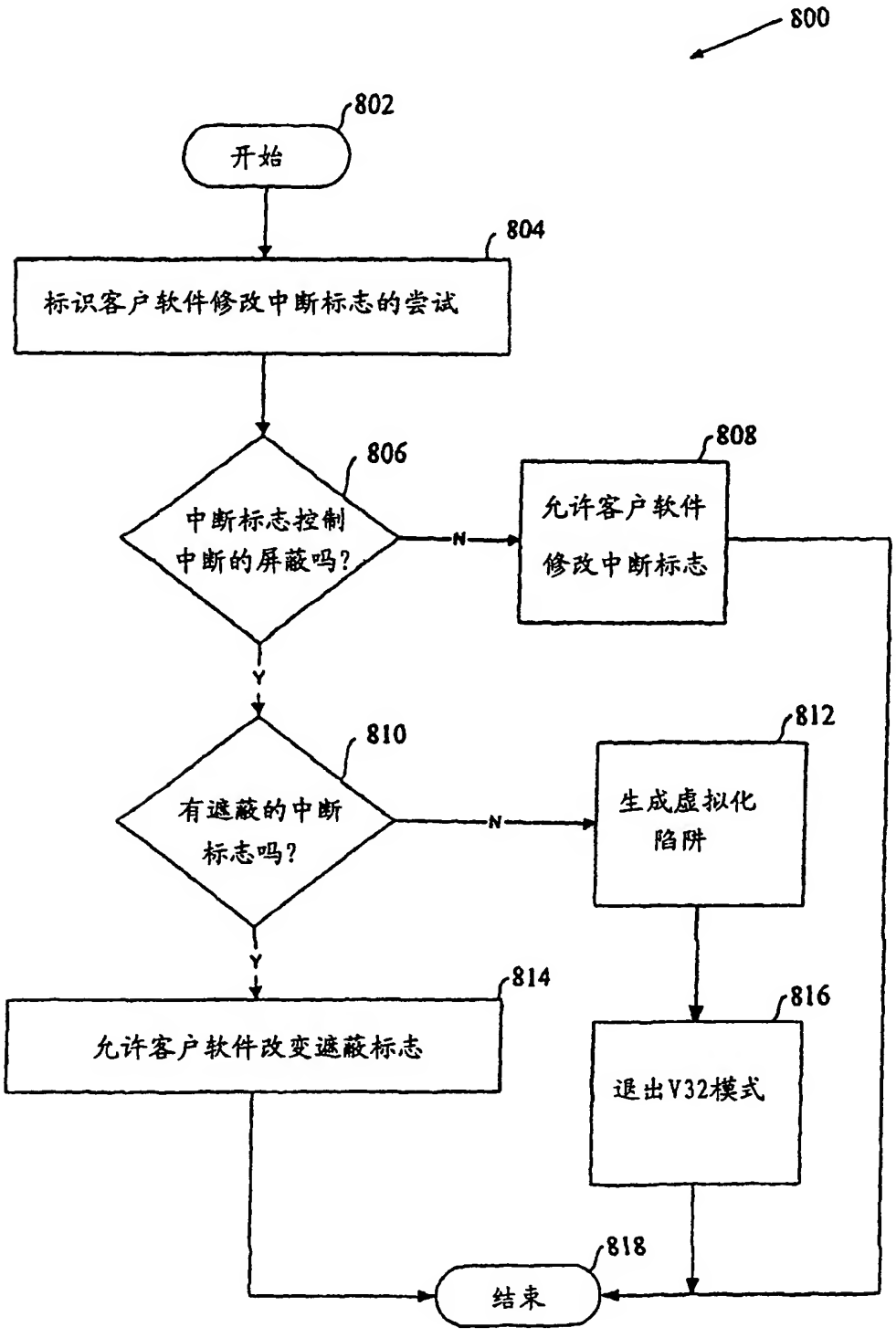

800
802
开始
804
标识客户软件修改中断标志的尝试
806
中断标志控制中断的屏蔽吗?
N
808
允许客户软件修改中断标志
Y
810
有遮蔽的中断标志吗?
N
812
生成虚拟化陷阱
Y
814
允许客户软件改变遮蔽标志
816
退出V32模式
818
结束

图 8

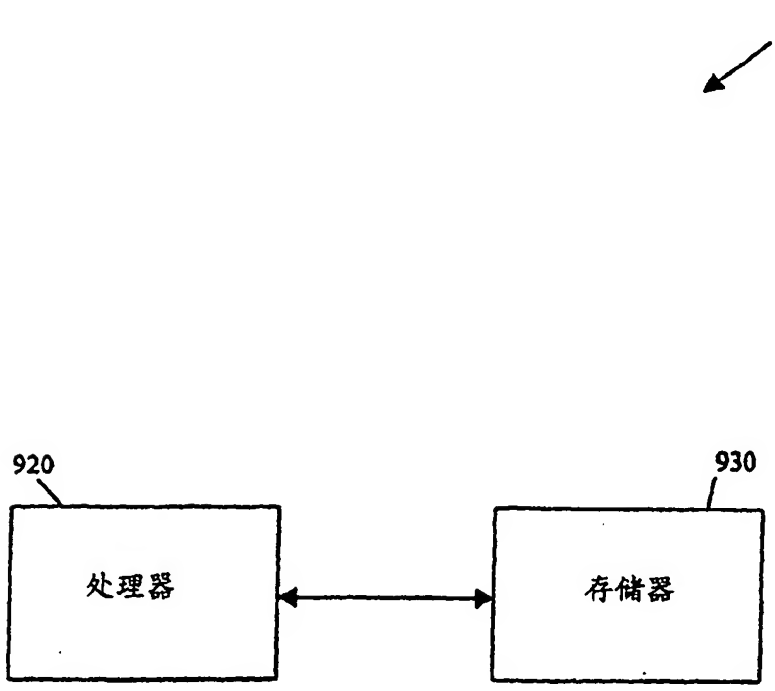


图 9